Plume Mapping via Hidden Markov Methods

Jay A. Farrell, *Senior Member, IEEE*, Shuo Pang, and Wei Li

***Abstract*—This paper addresses the problem of mapping likely locations of a chemical source using an autonomous vehicle operating in a fluid flow. The paper reviews biological plume-tracing concepts, reviews previous strategies for vehicle-based plume tracing, and presents a new plume mapping approach based on hidden Markov methods (HMMs). HMMs provide efficient algorithms for predicting the likelihood of odor detection versus position, the likelihood of source location versus position, the most likely path taken by the odor to a given location, and the path between two points most likely to result in odor detection. All four are useful for solving the odor source localization problem using an autonomous vehicle. The vehicle is assumed to be capable of detecting above threshold chemical concentration and sensing the fluid flow velocity at the vehicle location. The fluid flow is assumed to vary with space and time, and to have a high Reynolds number $(\mathrm{Re} > 10)$.**

***Index Terms*—Autonomous vehicles, hidden Markov methods (HMMs), online mapping, online planning, plume tracing.**

I. Introduction

OLFACTORY-BASED mechanisms have been hypothesized for a variety of biological behaviors [10], [39], [43]: homing by Pacific salmon [18]; homing by green sea turtles [25]; foraging by Antartic procellariiform seabirds [30], foraging by lobsters [1], [3], [9]; foraging by blue crabs [42]; and mate-seeking and foraging by insects [6], [7], [26]. Typically, olfactory-based mechanisms proposed for biological entities combine a large-scale orientation behavior based in part on olfaction with a multisensor local search in the vicinity of the source. The long-range olfactory-based search is documented in moths at ranges of 100 m–1000 m [12], [34] and in Antartic procellariiform seabirds over thousands of kilometers [30].

This paper considers the development of algorithms to replicate these feats in autonomous vehicles. The goal of the autonomous vehicle will be to locate the source of a chemical that is transported in a turbulent fluid flow. Because the ultimate intent is to implement these algorithms on autonomous vehicles, the computational efficiency of the resulting algorithms is a key concern. Such autonomous vehicle capabilities have applicability in searching for environmentally interesting phenomena, unexploded ordinance, undersea wreckage, and sources of hazardous chemicals or pollutants.

An initial approach to designing an autonomous vehicle plume-tracing strategy might attempt to calculate a concentration gradient, with subsequent plume tracing based on gradient following. Gradient-following-based plume tracing has been proposed for a few biological entities that operate in low Reynolds number environments [5]; however, gradient-based algorithms are not feasible in environments with medium to high Reynolds numbers [11], [21], and [28]. At low Reynolds numbers, the evolution of the chemical distribution in the flow is dominated by molecular diffusion and the concentration field is reasonably well defined by a continuous function with a peak near the source. At medium and high Reynolds numbers, the evolution of the chemical distribution in the flow is turbulence dominated [35]. The eddies of the turbulent advection process disperse the chemical by stretching and folding the chemical-containing parcels. The result of the turbulent diffusion process is a highly discontinuous and intermittent distribution of the chemical [21], [29].

If a dense array of sensors were distributed over an area, through which a turbulent flow was advecting a chemical, and the output of each sensor were averaged for a suitably long time (i.e., several minutes), then this average chemical distribution would be Gaussian [37], [38]. The required dense spatial sampling and long time-averaging, however, makes such an approach inefficient for implementation on a vehicle. In addition, only decameters from the odor source in the direction of the flow the gradient is too shallow to detect in a time-averaged plume. For an "instantaneous" plume, the gradient is time-varying, steep, frequently in the wrong direction, and would require numerous sensors. Therefore, gradient following is not practical.

It is known that the instantaneous odor distribution will be distinct from the time-averaged plume [21], [28]. The major differences include: the time-averaged plume is smooth and unimodal, while the instantaneous plume is discontinuous and multimodal; the time-averaged plume is time invariant (assuming ergodicity) while the instantaneous plume is time-varying; and, instantaneous concentrations well above the time-averaged concentration will be detected much more often than predicted by the Gaussian plume model. Such time-averaged plumes are useful for long-term exposure studies, but are not useful for studies of responses to instantaneously sensed odor [11], [28]. One of the reasons that olfaction is a useful long distance sensor is the fact that instantaneous concentrations well above the time-average are available at significant distances from the source [17]. The challenge for using olfaction on autonomous vehicles is to design effective algorithms to determine the odor source location even though

Manuscript received March 26, 2002. This work was supported by ONR Grant N00014-98-1-0820 from the DARPA/ONR Chemical Plume Tracing (CPT) Program, directed by K. Ward and R. Dugan, and by ONR Grant N00014-01-1-0906, from the ONR Chemical Sensing in the Marine Environment (CSME) Program, directed by K. Ward. This paper was recommended by Associate Editor M. S. de Queiroz.

J. Farrell and S. Pang are with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: farrell@ee.ucr.edu; spang@ee.ucr.edu).

W. Li is with the Department of Computer Science, California State University, Bakersfield, CA 93309 USA (e-mail: wli@cs.csubak.edu).

Digital Object Identifier 10.1109/TSMCB.2003.810873

the odor source concentration is not known, the advection distance of the detected odor is unknown, and the flow varies with both location and time.

Various studies have developed biomemetic robotic plume-tracing algorithms based on olfactory sensing. Belanger and Willis [4] presented plume-tracing strategies intended to mimic moth behavior and analyzed the performance in a "wind tunnel-type" computer simulation. The main goal of that study was to improve the understanding of moth interaction with an odor stimulus in a wind tunnel. Grasso *et al.* [15]–[17] evaluated biomimetic strategies and challenge theoretical assumptions of the strategies by implementing biomimetic strategies on their robot lobster. Robots that replicate biological approaches for plume tracing are also described in [19], [20], and [23]. Li *et al.* [24] developed, optimized, and evaluated a counter-turning strategy inspired by moth behavior. The fundamental aspects of these research efforts are sensing the chemical, sensing or estimating the fluid velocity, and generating a sequence of searcher speed and heading commands such that the motion is likely to locate the odor source. In each of these papers, the algorithms for generating speed and heading commands use only instantaneous (or very recent) sensor information. Typical orientation maneuvers include: sprinting upwind upon detection, moving crosswind when not detecting, and manipulating the relative orientation of a multiple sensor array to either follow an estimated plume edge or maintain the maximum mean reading near the central sensor.

Turbulent diffusion results in filaments of high concentration odor at significant distances from the source, but also results in high intermittency [2], [21], [27]–[29]. Intermittency increases with downflow distance both due to the meander of the instantaneous plume caused by spatial and temporal variations in the flow, and due to the increasing spread with distance of the filaments composing the instantaneous plume. High intermittency and large search areas motivate the need to acquire as much information as is possible from each odor detection event.

Engineered plume-tracing devices have sensing and computational capabilities that may not be available to biologicial entities. For example, an autonomous system may be able to record flow velocity and sensed concentration as a function of time and the vehicle position. Therefore, it is of interest to construct algorithms to effectively utilize these additional sensing and computational capabilities. This paper applies HMMs to the problem of odor source localization. This methodology results in algorithms for predicting likelihood of odor detection versus position, likelihood of source location versus position, the path most likely to have been taken by odor to a given location, and the path between two points that is most likely to detect odor.

The assumptions made herein relative to the chemical and flow are that the chemical is a neutrally buoyant and passive scalar being advected by a turbulent flow. The autonomous vehicle (or robot) is assumed to be capable of sensing position, concentration, and flow velocity. The concentration sensor is a binary detector. We analyze the plume mapping problem in two dimensions. A main motivation for implementing the mapping algorithms in two dimensions is the computational simplification achieved; however, neutral buoyancy of the chemical or stratification of the flow [36] will often result in a plume of limited vertical extent. Crawling insects and marine creatures restrict their odor source search to the bottom flow region. There is also evidence that moths stabilize altitude while tracing plumes [32]. The algorithms presented herein do extend directly to the three-dimensional (3-D) problem, but implementation for three dimensions requires significantly more computation.

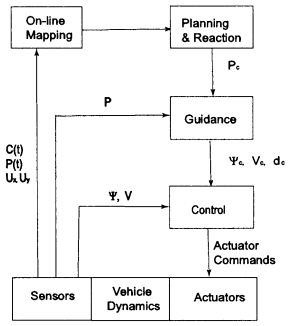


Fig. 1. Vehicle mapping, planning, and control architecture.

A table summarizing the notation used in this paper is given in Appendix III.

II. Autonomous Vehicle-Based Plume Tracing

The plume-tracing problem can be divided into two subproblems. First, assuming that at the ith time instant t_i a record of the flow velocity $(u_x(p_v(t_i)), u_y(p_v(t_i)))$ and concentration detection history $(c(p_v(t_i), t_i))$ at the vehicle location $p_v(t_i) = (x(t_i), y(t_i))$ is available, construct a map indicating which regions are likely to contain the odor source. Second, based on the source likelihood map (SLIM), plan paths that can accumulate information useful for improving the map, maximize the likelihood of the detection of odor, or maximize the likelihood of finding the source. Source likelihood mapping is useful for decreasing the time to find the source and in situations where, mission constraints require the vehicle to enforce a minimum standoff distance from the source. Also, even when the primary vehicle goal is to find the location of the source, if the vehicle fails to achieve this primary goal it is better to provide a SLIM than to return with no information. This paper focuses on the solution of these problems using HMMs [31], [33].

A typical vehicle hardware, control, guidance, mapping, and planning architecture is shown in Fig. 1. The figure shows that the assumed inputs to the online source likelihood mapping (OSLIM) system are sensed concentration $c(p_v(t_i), t_i)$, vehicle location $p_v(t_i)$, and flow velocity $u(t_i) = (u_x(p_v(t_i)), u_y(p_v(t_i)))$. The online planner would optimize a desired vehicle trajectory based on the OSLIM. The guidance system generates heading and speed commands to the controller to achieve the trajectory desired by the planner. This paper only considers algorithms useful to the OSLIM problem.

Assuming that odor is detected at $p_v(t_i)$, the basic idea of the online mapping algorithm is to use for $t_j \leq t_i$ the flow velocity record $\{u(p_v(t_j)t_j)\}_{j=0}^{i}$ and the detection record at the vehicle location $\{c(p_v(t_j), t_j)\}_{j=0}^{i}$ to estimate the likely previous trajectory of the chemical detected at $p_v(t_i)$. Accumulation of such odor trajectories across many detection events will allow construction of the OSLIM. Note that lack of odor detection can

be used similarly to decrease the OSLIM in appropriately defined regions. This paper presents computationally efficient algorithms for the required computations.

For example, if the flow velocity field $u(p,t)$ was known, where p denotes an arbitrary location in the search area, then the trajectory of the parcel detected at time t_i by the detector at location $p_v(t_i)$ could be calculated as

$$p_c(\tau) = p_v(t_i) + \int_{t_i}^{\tau} u(p_c(s), s)\, ds, \quad \tau \le t_i.$$

This backward integration calculation shows that the flow field is a function of both position and time. It also shows that the duration of integration $t_i - \tau$ is not known. When odor is detected at $p_v(t_i)$, the calculation provides a trajectory along which the source is located. When odor is not detected at $p(t)$, by a perfect sensor, the calculation provides a trajectory along which the source is not located.

The vehicle is not equipped with perfect detectors or with global flow velocity information. Olfactory sensing is characterized by very low false alarm rates, but potentially high missed detection rates. The high missed detection rate is due to the patchy distribution of chemical caused by turbulent diffusion. These stochastic factors must be accounted for in the mapping algorithms. The uncertainty in $u(p_c(s), s)$ for $s \le t$, especially since only $\{u(p_v(t_j), t_j)\}_{j=0}^{i}$ is available, results in increasing uncertainty as the duration of the backward time integration increases. The likelihood mapping algorithms must account also for this distribution of possible trajectories from the source to the detector. For on-vehicle implementation this algorithm most be carefully constructed for computational feasibility. HMMs are manipulated herein to produce such algorithms.

III. Model Representation

A. Flow Velocity Sensor Processing

Mapping and planning algorithms compute at a lower rate than guidance and control algorithms. Therefore, in typical applications, there are M sensor readings per mapping algorithm update interval. The mapping algorithm will use the mean flow vector over these M measurements

$$u_x(t_i) = \frac{1}{M}\sum_{k=1}^{M} u_x(t_{i-1} + kdt) \tag{1}$$

$$u_y(t_i) = \frac{1}{M}\sum_{k=1}^{M} u_y(t_{i-1} + kdt) \tag{2}$$

where $dt = (t_i - t_{i-1}/M)$. Note that for notational convenience, we have dropped the explicit representation of flow as a function of position. All measurements occur at the location of the sensor on the vehicle. The mapping algorithm will use the peak concentration measurement over the M concentration measurements [i.e., $c(t_i) = \max_{k=1,\ldots,M}(c(t_{i-1} + kdt))$]. Because detection events are rare, this ensures that no detection events are missed.

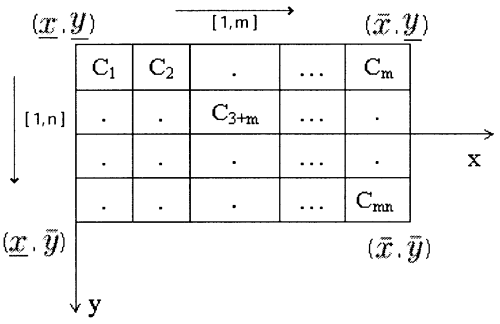


Fig. 2. Cellular subdivision of the region to be searched.

B. Plume Map Representation

Both for computational feasibility, and to construct a model suitable for the HMM approach, a rectangular region is defined that covers the search area that is of interest. A set of coordinates for and an $n \times m$ cellular subdivision of this rectangular area is defined as shown in Fig. 2.

Define a vector of cells $C = [C_1, \ldots, C_N]$ that covers the area of interest, where $N = mn$. Let $i \in [1, m]$ count over cells in the x direction. Let $j \in [1, n]$ count over cells in the y direction. Knowledge of the vehicle position allows direct calculation of the (i, j) indexes of the cell containing the vehicle. Given i and j, the index of the cell element is $k = i + (j-1)m$. The inverse mapping from k to (i, j) is

$$i(k) = \mathrm{rem}(k-1, m) + 1 \tag{3}$$

$$j(k) = \mathrm{int}\left(\frac{k-1}{m}\right) + 1 \tag{4}$$

where $\mathrm{int}(n, m)$ is the greatest integer less than or equal to (n/m), and $\mathrm{rem}(n, m)$ is the remainder of n divided by m. Therefore, the notations $C_{i,j}$ and C_k are equivalent. In addition, one additional cell C_0 is introduced. As will be shown later, this extra cell simplifies some later computations and normalizations. This cell can be conceptualized as the environment outside the search area. Therefore, when odor leaves the search area, it enters cell C_0.

Let $0 \le \pi_k \le 1$ represent the probability that there is an odor source in C_k. The vector $\pi = [\pi_1, \ldots, \pi_N]$ is initially unknown. This N vector can be converted to an $n \times m$ array and interpreted as the OSLIM. This map is one of the items that we will be attempting to estimate. Note that if it is assumed that there is exactly one source in the region of interest, then $\sum_{k=1}^{N} \pi_k = 1$. Initially, if there is no prior information about the source location, then π is initialized uniformly as $\pi_i = (1/N)$. The π vector can be initialized nonuniformly, if prior information about source location is known.

C. Hidden Markov Plume Model

The hidden Markov plume model (HMPM) is represented by the parameter vector $\lambda = [\pi, \{\mathbf{A}(t_i)\}_{i=0}^{f}, \mathbf{b}]$ where π is the source probability vector (see Section III-B), $\mathbf{A}$ is the state transition matrix, and $\mathbf{b}$ is the detection probability vector.

Let $a_{kl}(t_i)$ represent the probability of the transition of detectable odor from $C_k(t_i)$ to $C_l(t_{i+1})$. Then

$$\mathbf{A}(t_i) = [a_{kl}(t_i)] \in \Re^{(N+1)\times(N+1)} \tag{5}$$

is the matrix of cell transition probabilities at time t_i. Since π represents the source probability vector, if we define $a_{0j}(t_i) = \pi_j$, then $a_{0j}(t_i)$ represents the probability that odor from the source released at t_i is in cell C_j at time t_i. Furthermore, we interpret $a_{k0}(t_i)$ as the probability that odor from cell C_k leaves the search region at time t_i. Note[1] that since all odor in $C_k(t_i)$ must go someplace at t_{i+1}, we have the constraint that $\sum_{l=0}^{N} a_{kl}(t_i) = 1$. The definition of $\mathbf{A}(t_i)$ based on $\mathbf{u}(t_i)$ is given in Appendix II. For certain computations, the zeroth row and column will not be important; therefore, we define $\bar{\mathbf{A}}(t_i) = [a_{kl}(t_i)] \in \Re^{N\times N}$ for $k, l \in [1, N]$.

The detection probability vector $\mathbf{b}$ is the probability of detecting odor in each cell if there is detectable odor in that cell. Since the sensor performance is assumed to be independent of the sensor location, the elements of b are identical and can be represented by a known constant μ times a unity vector. The probability of detecting odor in C_k at time t_i is therefore $b_k\alpha_k(t_i) = \mu\alpha_k(t_i)$, where $\alpha_k(t_i)$ represents the probability of cell k containing detectable odor at time t_i. An efficient algorithm for calculation of $\alpha_k(t_i)$ is presented in Section IV-A.

Corresponding to the traditional HMM literature, three problems are of interest.

1) Use the model λ to predict $P_\lambda(O)$, where $P_\lambda(O)$ represents the probability of the observed set of concentration detection events denoted by O.
2) Use the model λ to estimate the state sequence S that yielded the observations O.
3) If λ is not known, then find the model $\hat{\lambda}$ that maximizes $P_{\hat{\lambda}}(O)$. Since b is known and a_{ij} for $i \in [1, N]$ and $j \in [0, N]$ can be computed from the flow velocity history, the main issue is the estimation of π (or a_{0j} for $j \in [1, N]$).

Letting $D(p_v(t_i)) \in \{0, 1\}$ represent *detection* (i.e., $D = 1$) and *no detection* (i.e., $D = 0$) events at the vehicle location p_v at time t_i. Then, the observation vector at time t is

$$O = [D(p_v(t_1)), \ldots, D(p_v(t_i))].$$

The sequence of cells most likely to have been transitioned by the odor to result in the detection event $D(p_v(t_i))$ is denoted by $S(D(p_v(t_i)))$. Determination of $S(D(p_v(t_i)))$ is a stochastic extension of the backward integration discussed in Section II. The appropriate algorithm is presented in Section IV-C2.

IV. Hidden Markov-Based Tools

This section adapts methods from HMM [31], [33] to the solution of important questions applicable to developing an OSLIM and to defining useful trajectories related to the plume tracing and source localization problems. Section IV-A addresses the first problem stated in Section III-C: how to use the model λ to predict $P_\lambda(D(p_v(t_i)))$. Section IV-B presents an algorithm for calculating the likelihood of a source in cell i producing odor that is detected in cell k at time t_L. Section IV-B1 presents an algorithm for estimation of the unknown quantity π of the HMM model $\hat{\lambda}$. Section IV-C2 presents an algorithm for determining, given $\hat{\lambda}$, the most likely path that odor would have taken between two cells at two given times. Section IV-C3 presents an algorithm for determining the connected path between two cells that is most likely to detect odor.

[1]This constraint holds for a 3-D problem even for a two-dimensional (2-D) implementation. For example, the cells can be considered as having either fixed height. Odor leaving the vertical edges has then left the search area (i.e., entered C_0).

A. Plume Location Likelihood Map

The probability of each detection event is (by Baye's rule for conditional probabilites)

$$P(D(p_v(t_i))) = \mu\alpha_k(t_i, t_0) \tag{6}$$

where

- C_k cell containing $p_v(t_i)$;
- M detection probability given that the cell contains detectable odor;
- $\alpha_k(t_i, t_0)$ probability that C_k contains detectable odor at time t_i due to the continuous release of odor by the source starting at t_0.

Since μ is a known fixed constant, the key issue is calculation of $\alpha_k(\tau, t_0), \tau \in [t_0, t_i]$.

Introduce the intermediate variable $\bar{\alpha}_k(t_i, t_o)$ that represents the probability that C_k contains detectable odor at time $t_i \geq t_o$ due to an odor release only at time t_o. Let

$$\bar{\alpha}(t_i, t_o) = [\bar{\alpha}_1(t_i, t_o), \ldots, \bar{\alpha}_N(t_i, t_o)]$$

be the vector storing this variable for each cell. Since π is the source probability vector, $\bar{\alpha}(t_o, t_o) = \pi$. The calculation of $\bar{\alpha}_l(t_1, t_o)$ must account for the transition probability from all other cells to cell l. Therefore, $\bar{\alpha}_l(t_1, t_o) = \sum_{k=1}^{N} \bar{\alpha}_k(t_o, t_o)a_{kl}(t_o)$. In vector notation, $\bar{\alpha}(t_1, t_o) = \bar{\alpha}(t_o, t_o)\bar{\mathbf{A}}(t_o)$ and

$$\bar{\alpha}(t_n, t_o) = \begin{cases} 0, & \text{for } t_n < t_o \\ \pi\mathbf{I}, & \text{for } t_n = t_o \\ \pi\prod_{j=0}^{n-1} \bar{\mathbf{A}}(t_j), & \text{for } t_n > t_0 \end{cases} \tag{7}$$

where $\prod_{j=0}^{n} \bar{\mathbf{A}}(t_j) = \bar{\mathbf{A}}(t_o)\ldots\bar{\mathbf{A}}(t_n)$. Let $\Phi(t_{n+1}, t_o) = \prod_{j=0}^{n} \bar{\mathbf{A}}(t_j)$. The computational of $\Phi(t_{n+1}, t_o)$ for the case $t_n > t_0$ requires $n$matrix multiples of dimension $N \times N$, which requires $O(nN^3)$ Floating point OPerationS (FLOPS) at each step. Alternatively, $\Phi(t_{n+1}, t_o)$ can be calculated as

$$\Phi(t_n, t_o) = \Phi(t_{n-1}, t_o)\bar{\mathbf{A}}(t_{n-1})$$

which requires $O(N^3)$ FLOPS per time update. The computation of $\bar{\alpha}(t_n, t_o)$ by (7) can be rewritten in either of the following recursive formulations:

$$\bar{\alpha}(t_n, t_o) = \bar{\alpha}(t_{n-1}, t_o)\bar{\mathbf{A}}(t_{n-1}), \quad \text{for } t_n > t_o \tag{8}$$

or

$$\bar{\alpha}(t_n, t_o) = \pi\Phi(t_n, t_o). \tag{9}$$

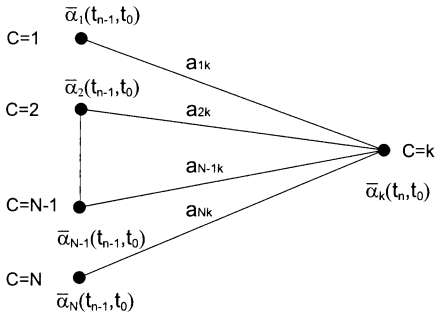


Fig. 3. One forward propagation step for the plume map $\bar{\alpha}_k(t)$ by (8) where a_{ik} is the transition probability from cell i to cell k.

The computation of (8) is illustrated in Fig. 3. Equation (8) requires $O(N^2)$ FLOPS per time update. Due to the update of Φ, (9) requires $O(N^3)$ FLOPS per time update.

The variable $\bar{\alpha}_k(t_n, t_o)$ accounts for the transport of the odor released at the single instant t_o. The variable $\alpha_k(t_n, t_o)$ that accounts for a continuous release of odor from t_o to time t_n is then calculated as

$$\alpha(t_n, t_o) = \frac{1}{n+1}\left(\sum_{j=0}^{n} \bar{\alpha}(t_n, t_j)\right) \tag{10}$$

where the factor of $(n+1)$ is introduced to maintain $\|\alpha(t_n, t_o)\|_1 = 1$. This expression reduces, using (8), as

$$\begin{aligned}\alpha(t_n, t_o) &= \frac{1}{n+1}\left[\pi I + \sum_{j=0}^{n-1}(\bar{\alpha}(t_{n-1}, t_j)\bar{\mathbf{A}}(t_{n-1}))\right] \\ &= \frac{1}{n+1}\left[\pi I + \left(\sum_{j=0}^{n-1}\bar{\alpha}(t_{n-1}, t_j)\right)\bar{\mathbf{A}}(t_{n-1})\right] \\ &= \frac{1}{n+1}[\pi I + n\alpha(t_{n-1}, t_o)\bar{\mathbf{A}}(t_{n-1})] \end{aligned} \tag{11}$$

for $n \geq 0$. This algorithm is summarized in Table I. Since this recursive relation depends only on the most recent $\bar{\mathbf{A}}$ matrix, the past $\bar{\mathbf{A}}$ matrices would not need to be stored. This recursive computation of $\alpha(t_n, t_o)$ by (11) requires $O(N^2)$ FLOPS per time step.

The variable $\alpha(t_n, t_o)$ could also be calculated using (9) as

$$\begin{aligned}\alpha(t_n, t_o) &= \frac{1}{n+1}\left[\sum_{j=0}^{n}\pi\Phi(t_n, t_j)\right] \\ &= \frac{\pi}{n+1}\left[\sum_{j=0}^{n}\Phi(t_n, t_j)\right] \\ &= \frac{\pi}{n+1}\left[I + \sum_{j=0}^{n-1}\Phi(t_n, t_j)\right] \\ &= \pi\Psi(t_n, t_o) \end{aligned} \tag{12}$$

where $\Psi(t_n, t_o) = (1/n+1)[I + \sum_{j=0}^{n-1}\Phi(t_n, t_j)]$, which satisfies the recursion relation

$$\Psi(t_n, t_o) = \frac{1}{n+1}[n\Psi(t_{n-1}, t_o)\bar{\mathbf{A}}(t_{n-1}) + \mathbf{I}]. \tag{13}$$

TABLE I
ALGORITHM TO GENERATE A PLUME LIKELIHOOD MAP FOR THE HYPOTHESIZED SOURCE LIKELIHOOD VECTOR π, WHERE $\alpha_k(t_n, t_o)$ DENOTES THE PROBABILITY OF THERE BEING DETECTABLE ODOR IN CELL k AT TIME t_n DUE TO CONTINUOUS RELEASE FOR $t \in [t_o, t_n]$ AND a_{ik} ARE ELEMENTS OF THE STATE TRANSITION MATRIX DETERMINED BY THE FLOW

1. **Initialize:** $\alpha_i(t_o, t_o) = \pi_i$ for $i = 0, \ldots, N$.

2. **Induction:**

$$\alpha_k(t_n, t_o) = \frac{1}{n+1}\left(\pi_k + n\sum_{i=1}^{N}\alpha_i(t_{n-1}, t_o)a_{ik}(t_{n-1})\right)$$

for $k = 1, \ldots, N$.

3. **Termination:** $Pr(C_k(t_n, t_o)|\lambda) = b(k)\alpha_k(t_n, t_o)$

The ith row of $\Psi(t_n)$ represents the probability at t_n, given a continuous release of odor starting at t_o, from a source in C_i that there is detectable odor in any other cell. Since this recursive relation depends only on the most recent $\bar{\mathbf{A}}$ matrix, the past $\bar{\mathbf{A}}$ matrices do not need to be stored. The recursive computation of $\alpha(t_n, t_o)$ by (12) requires $O(N^2)$ FLOPS per time step. The recursive computation of $\Psi(t_n, t_o)$ by (13) requires $O(N^3)$ FLOPS per time step.

The algorithms of (11) and (12) are forward calculations that, given the model λ, project the probability of odor being in any cell. For (11), the first iteration initializes the probability $\alpha(t_o, t_o)$ based on the hypothesized probability vector represented by π. Subsequent iterations calculate the probabilities at time t_{n+1} based on the probabilities at time t_n and the transition probabilities $a_{ij}(t_n)$. By embedding the vector π in the computation at each time, the algorithm of (11) is able to be implemented with significantly fewer computations than are required for the algorithm of (12). The savings is the result of implementing a vector-matrix product instead of the matrix-matrix product necessary in (13).

In spite of the fact that the algorithm of (12) requires additional computation, the form of (12) is important, since in the model λ, π is the only unknown. The parameter $\mathbf{A}(t_n)$ is calculated based on the fluid flow. Therefore, (12) allows prediction of the plume likelihood map (probability that each cell contains detectable odor) that would result from any hypothesized source probability vector π. The algorithm of (11) would require complete recalculation from t_o to the present time; although (12) has a higher per time step computational load, it may have a lower computational load when computations will be required for different hypothesized values of π. Interpreting $\alpha_k(t_n, t_o)$ as a plume likelihood map calculated for the current estimate of π, allows a planner to construct trajectories based on maximizing the likelihood of contacting the plume that would result from a hypothesized π.

B. Odor Path Likelihood Map

It will be useful to have an algorithm to predict the probability, given the flow history, that a source in any given cell has transported odor to $C_k(t_L)$. To this end, let $\beta_{ij}(t_L, t_F)$ denote

TABLE II
ALGORITHM TO GENERATE A MAP $\beta_{ij}(t_L, t_F)$ OF THE LIKELIHOOD OF CELL i GENERATING ODOR AT TIME t_F THAT TRANSITIONS TO CELL j AT TIME t_L, WHERE a_{ik} ARE ELEMENTS OF THE STATE TRANSITION MATRIX DETERMINED BY THE FLOW

1. **Initialize:** $\beta_{ij}(t_L, t_L) = \delta_{ij}$ where δ_{ij} is the Kronecker delta function and j denotes the cell of interest.

2. **Induction:**

$$\beta_{ij}(t_L, t_{f-1}) = \sum_{k=1}^{N} a_{ik}(t_{f-1})\beta_{kj}(t_L, t_f), \ f \in [F+1, L].$$

3. **Termination**

(a) $\beta_{ij}(t_L, t_F)$ is the probability that odor released in $C_i(t_F)$ transitions to $C_j(t_L)$. The vector $\beta_{ij}(t_L, t_F)$ for j fixed represents a map of which cells are likely to have transported odor released at t_F to $C_j(t_L)$.

(b) Alternatively, $\pi\beta_{ij}(t_L, t_F)$ is the probability of odor transitioning to $C_k(t_L)$ when the odor sources have the distribution π.

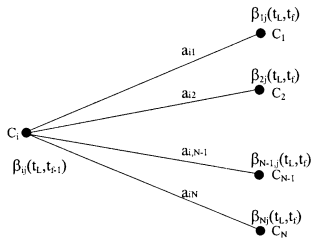


Fig. 4. Graphical illustration for the derivation of the odor path likelihood map $\beta_{ij}(t_L, t_{f-1})$ by the backward propagation algorithm of (14) where a_{ik} is the transition probability from cell i to cell k.

the probability that odor released in $C_i(t_F)$ transitions to $C_j(t_L)$ for $t_L > t_F$. With this definition, $\beta(t_L, t_L)$ is initialized as

$$\beta_{ij}(t_L, t_L) = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$$

where $i \in [1, N]$ and for the present time j is fixed. The desired algorithm is derived below based on the HMM backward propagation algorithm of [33, Sect. III-A].

The parameter $\beta_{ij}(t_L, t_F)$ can be calculated by the algorithm in Table II. To understand this algorithm, consider Fig. 4. For $t_F = t_{L-1}$, the probability of odor transitioning from $C_i(t_{L-1})$ to $C_k(t_L)$ is simply $\beta_{ik}(t_L, t_{L-1}) = a_{ik}(t_{L-1})$ since there is a single transition path. Since $\beta_{ik}(t_L, t_L) = \delta_{ik}$, this can also be expressed as $\beta_{ik}(t_L, t_{L-1}) = a_{ik}(t_{L-1}) = \sum_{j=0}^{N} a_{ij}(t_{L-1})\beta_{jk}(t_L, t_L)$. For $t_F < t_{L-1}$, the probability of odor transitioning from $C_i(t_F)$ to $C_k(t_L)$ must account for all possible transition sequences from cell i to cell k in $L - F$ steps. In spite of the huge number of possible paths (i.e., $(N)^{L-F}$), HMM methods provide a convenient algorithm. At time t, there are N possible next cells. By Bayesian arguments

$$\begin{aligned}\beta_{ij}(t_L, t_F) &= \sum_{k=1}^{N} \beta_{ik}(t_{F+1}, t_F)\beta_{kj}(t_L, t_{F+1}) \\ &= \sum_{k=1}^{N} a_{ik}(t_F)\beta_{kj}(t_L, t_{F+1}).\end{aligned}$$

Let $\beta_j(t_L, t_F) = [\beta_{1j}(t_L, t_F), \ldots, \beta_{Nj}(t_L, t_F)]^T$. Since $\beta_j(t_L, t_L)$ is known, $\beta_j(t_L, t_F)$ can be propagated backward through time for any $t_F < t_L$ as

$$\beta_j(t_L, t_F) = \bar{\mathbf{A}}(t_F)\beta_j(t_L, t_{F+1}). \tag{14}$$

For a fixed t_L, this propagation of $\beta_j(t_L, t_F)$ backward to time t_F requires all $\bar{\mathbf{A}}(t_k)$ for $k \in [F, L)$ to be available. Computation of $\beta_j(t_L, t_F)$ would require $L - F$ vector-matrix multiplies (i.e., $(L - F)N^2$ FLOPS) per time step. This process would be repeated at t_{L+1} requiring $L - F + 1$ vector-matrix multiplies. Therefore, as written the algorithm has computational and memory requirements that grow with time. Note however that

$$\begin{aligned}\beta_j(t_L, t_F) &= \bar{\mathbf{A}}(t_F)\beta_j(t_L, t_{F+1}) \\ &= \bar{\mathbf{A}}(t_F)\bar{\mathbf{A}}(t_{F+1})\cdots\bar{\mathbf{A}}(t_{L-1})\beta_j(t_L, t_L) \\ &= \Phi(t_L, t_F)\beta_j(t_L, t_L)\end{aligned}$$

where the transition matrix $\Phi(t_L, t_F)$ was defined after (7). If $\Phi(t_{L-1}, t_F)$ is available, then $\Phi(t_L, t_F) = \Phi(t_{L-1}, t_F)\bar{\mathbf{A}}(t_{L-1})$; therefore, $\Phi(t_{L-1}, t_F)$ can be propagated forward in time as each new $\bar{\mathbf{A}}(t_L)$ becomes available. Then, the likelihood of each cell at time t_F propagating detectable odor to cell j at time t_{L+1} is updated as

$$\beta_j(t_{L+1}, t_F) = \Phi(t_{L+1}, t_F)\beta_j(t_{L+1}, t_{L+1}) \tag{15}$$

where $\beta_j(t_{L+1}, t_{L+1}) = \Delta_j = [\delta_{1j}, \ldots, \delta_{Nj}]^T$ is trivial to define when j is known and $\Phi(t_{L+1}, t_F)$ is updated based on $\Phi(t_L, t_F)$ and $\bar{\mathbf{A}}(t_L)$. The update of (15) has fixed memory and computational requirements.

Therefore, by maintaining the state transition matrix $\Phi(t_{L+1}, t_F)$, we have that the ith row of $\Phi(t_{L+1}, t_F)$ represents a map of which cells are likely to contain detectable odor if odor were released in $C_i(t_F)$. The jth column of $\Phi(t_{L+1}, t_F)$ represents a map of the likelihood of each cell containing the source that release odor at t_F that was transported to $C_j(t_L)$. An important advantage of maintaining $\Phi(t_{L+1}, t_F)$ is that different rows or columns are available as they may be needed without any recomputation.

Computation of $\beta_{ik}(t_L, t_F)$ accounts only for transitions from cell i at time t_F to cell k at time t_L. Because we do not know the propagation time $t_L - t_F$ (i.e., we do not know the time t_F at which the detected chemical was released), we must account for all possible release times by defining

$$\bar{\beta}_{kj}(t_L, t_0) = \frac{1}{L+1}\sum_{i=0}^{L} \beta_{kj}(t_L, t_i) \tag{16}$$

where t_0 is the first time that data was available, $k = 1, \ldots, N$, and j is fixed. The parameter L is the number of backward time propagation steps. Selection of this parameter is constrained by computational load and the duration of time for which the flow velocity vector is available. Equation (16) is not efficient

in terms of computation or memory. A recursive version of this algorithm is developed as follows:

$$\begin{aligned}\bar{\beta}_j(t_L,t_0) &= \frac{1}{L+1}\sum_{i=0}^{L}\beta_j(t_L,t_i)\\ &= \frac{1}{L+1}\sum_{i=0}^{L}[\Phi(t_L,t_i)\beta_j(t_L,t_L)]\\ &= \frac{1}{L+1}\left(\sum_{i=0}^{L}\Phi(t_L,t_i)\right)\beta_j(t_L,t_L)\\ &= \Psi(t_L,t_0)\beta_j(t_L,t_L)\end{aligned}$$

where $\Psi(t_L,t_0)$ is propagated recursively by (13).

Therefore, by maintaining the superposition matrix $\Psi(t_L,t_0)$, we have that the ith row of $\Psi(t_L,t_0)$ represents a map of which cells are likely to contain detectable odor if odor were released continuously in $C_i(t_f)$ for $f \in [0,L]$. The jth column of $\Psi(t_L,t_0)$ represents a map of the likelihood of each cell containing a source whose continuous release would result in odor being transported to $C_j(t_L)$. An important advantage of maintaining $\Psi(t_L,t_0)$ is that different rows or columns (representing different source and destination locations) are available as they may be needed, without any recomputation. The disadvantage is the amount of computation required to maintain Ψ.

1) Source Likelihood Map (SLIM): If odor is detected in C_k at $t=t_L$, then $\bar{\beta}_{ik}(t_L,t_0)$ for $i \in [1,N]$ indicates which cells are likely to have contained the source that resulted in the detected odor in $C_k(t_L)$. The variable $\bar{\beta}_{ik}(t_L,t_0)$ can therefore be useful for adaptation of the source probability vector π. Similarly, if odor is not detected in C_k at time t_L, then $\bar{\beta}_{ik}(t_L,t_0)$ indicates which cells are unlikely to contain the source. In the following, $\bar{\beta}_k(t_L)=[\bar{\beta}_{1k}(t_L,t_0),\ldots,\bar{\beta}_{Nk}(t_L,t_0)]^T$.

Using these ideas, we adapt the estimate of the source likelihood vector $\hat{\pi}$ as follows. Assuming that no prior information is available about the location of the source in the search area, we initialize $\hat{\pi}$ uniformly over the region as $\hat{\pi}=[1,\ldots,1]/N \in \Re^N$. The update of $\hat{\pi}$ is defined as

$$\hat{\pi}(t_L)=\begin{cases}(1-\epsilon_d)\hat{\pi}(t_{L-1})+\epsilon_d\dfrac{\bar{\beta}_k(t_L)}{\|\bar{\beta}_k(t_L)\|_1}\\ \qquad \text{when odor is detected in } C_k(t_L)\\ (1+\epsilon_c)\hat{\pi}(t_{L-1})-\epsilon_c\dfrac{\bar{\beta}_k(t_L)}{\|\bar{\beta}_k(t_L)\|_1}\\ \qquad \text{when odor is not detected in } C_k(t_L).\end{cases}$$

In the detection case, if $\hat{\pi}(t_{L-1})$ is a probability vector, then $\hat{\pi}(t_L)$ will be a probability vector (i.e., $\|\hat{\pi}(t_L)\|_1=1$). In the latter (no detection) case, $\hat{\pi}(t_L)$ must be normalized so that its one norm again has magnitude one and each element of $\hat{\pi}(t_L)$ is in $[0,1]$. The design parameters of this algorithm are ϵ_c and ϵ_d. In the case where the probability of missed detection is high, then ϵ_c should be small. Both parameters must be positive with magnitude less than one.

It is important to note the distinction between $\pi(t_L)$ and $\bar{\beta}_k(t_L)$. The vector $\bar{\beta}_k(t_L)$ keeps track of the credit each cell deserves for a detection/no detection event in C_k at time t_L. The vector $\hat{\pi}(t_L)$ accumulates the $\bar{\beta}_k(t_i)$ information across all detection/no detection events (i.e., for $i \in [0,L]$) to estimate the likelihood that each cell contains the source.

TABLE III
VITERBI ALGORITHM WHERE $\delta_j(t_k)$ IS THE PROBABILITY OF THE MOST LIKELY CELL SEQUENCE TO C_j AT TIME t_k, a_{ij} ARE ELEMENTS OF THE STATE TRANSITION MATRIX DETERMINED BY THE FLOW, AND $\psi_j(t_k)$ IS THE INDEX OF THE MOSTLY LIKELY CELL TO TRANSPORT ODOR TO CELL j AT TIME t_k

1. **Initialize:** $\delta_i(t_0)=\pi_i$, $1 \le i \le N$ and $\psi_i(t_0)=0$.

2. **Recursion:**

$$\begin{aligned}\delta_j(t_k) &= \max_{1\le i\le N}[\delta_i(t_{k-1})a_{ij}(t_{k-1})],\\ &\qquad 1\le j\le N, 1\le k\le L\\ \psi_j(t_k) &= \arg\max_{1\le i\le N}[\delta_i(t_{k-1})a_{ij}(t_{k-1})]\end{aligned}$$

3. **Termination:** $q_j^*(t_L)=j$, $q_j^*(t_k)=\psi_{q_j^*(t_{k+1})}(t_{k+1})$

C. Most Likely Paths

The Viterbi algorithm (VA) [14], [40] can be adapted to generate paths through the cell space that are useful to the source localization problem. Section IV-C1 reviews the VA. Section IV-C2 adapts the VA to compute the most like path taken by odor between cells $C_i(t_F)$ and $C_j(t_L)$. Section IV-C3 adapts the VA to compute the connected path between cells $C_i(t_L)$ and $C_j(t_L)$ that is most likely to result in odor detection. In a time-varying flow field, these two paths are distinct.

1) Viterbi Algorithm (VA): The VA is a recursive, optimal solution to the problem of estimating the state sequence of a discrete-time, finite-state Markov process observed in memoryless noise. In its most general form, the VA may be viewed as a solution to the problem of maximum *a posteriori* (MAP) probability estimation of the state sequence between two states of a finite-state discrete-time system. The VA is summarized in Table III where $\delta_j(t_k)$ is the probability, given π at t_0, of the most likely cell sequence to $C_j(t_k)$ at time t_k; $\psi_j(t_k)$ is the index of the most like cell transitioning to $C_j(t_k)$; and $q_j^*=[q_j^*(t_L),q_j^*(t_{L-1}),\ldots,q_j^*(t_1)]$ is the most likely cell sequence to $C_j(t_L)$. The first step initializes the probability of the most likely cell sequence based on π. Step 2 calculates the probability of the most likely cell transition to cell j at time t_k based on the probability of the most likely cell sequences to each cell at time t_{k-1} and the cell transition probabilities at t_{k-1}. At the same time, we use $\psi_j(t_k)$ to record the cell number, which is the mostly likely cell to transport odor to cell j at time t_k.

Consider the following simple example of a three-state application of the VA to a generic (nonplume tracing) application. Assume that $\pi=[1,0,0]$

$$\mathbf{A}(0)=\begin{bmatrix}0.60 & 0.40 & 0.00\\ 0.00 & 1.00 & 0.00\\ 0.00 & 1.00 & 0.00\end{bmatrix}$$

and

$$\mathbf{A}(2)=\mathbf{A}(1)=\begin{bmatrix}0.25 & 0.25 & 0.50\\ 0.00 & 1.00 & 0.00\\ 0.00 & 0.50 & 0.50\end{bmatrix}.$$

At time $t=1, \delta(1)=[0.60,0.40,0.00]$ and $\psi(1)=[1,1,np]$ where np stands for not possible. At time $t=2, \delta(2)=[0.15,0.40,0.30]$ and $\psi(2)=[1,2,1]$. Therefore, the most

likely state sequence to $S_2(2)$ is $S_1(0) \rightarrow S_2(1) \rightarrow S_2(2)$. At time $t = 3, \delta(3) = [0.04, 0.40, 0.15]$ and $\psi(3) = [1, 2, 3]$. Therefore, the most likely state sequence to $S_3(3)$ is $S_1(0) \rightarrow S_1(1) \rightarrow S_3(2) \rightarrow S_3(3)$.

2) Most Likely Odor Path Between $C_i(t_s)$ and $C_j(t_f)$: For the plume-tracing application, direct application of the VA, using $\mathbf{A}(t_s)$ through $\mathbf{A}(t_f)$ calculated using $u = [u(t_s), \ldots, u(t_f)]$, generates the most likely odor path (i.e., cell sequence) to any desired final cell location (i.e., $C_j(t_f)$) for an assumed π. If $\hat{\pi}$ is used then the resulting cell sequence accounts for odor sources in all cells with the $\hat{\pi}$ vector appropriately weighting each cell.

Alternatively, if π for the VAs is defined to be zero in all cells except for being 1.0 in cell $C_i(t_s)$, then the resulting sequence is the most likely odor path (i.e., cell sequence) between the specified start location $C_i(t_s)$ and the end location $C_j(t_f)$.

The VA finds the most likely cell sequence forward through time. This cell sequence could also be calculated backward through time according to the following.

1) **Initialize:** $\psi_m(t_f) = 0$ and $\mu_m(t_f) = \delta_{mj}, 1 \leq m \leq N$ where δ_{mj} is the Kronecker Delta.
2) **Recursion:**

$$\mu_m(t_k) = \max_{1 \leq n \leq N} [a_{mn}(t_k)\mu_n(t_{k+1})] \qquad 1 \leq n \leq N, s+1 \leq k \leq f-1$$
$$\psi_m(t_k) = \arg \max_{1 \leq n \leq N} [a_{mn}(t_k)\mu_n(t_{k+1})].$$

3) **Termination:**

$$q_i^*(t_s) = i, \quad q_i^*(t_k) = \psi_{q_i^*(t_{k-1})}(t_{k-1})$$
$$\mu_m(t_s) = \max_{1 \leq n \leq N} [\pi_m a_{mn}(t_s)\mu_n(t_1)] \tag{17}$$

where $\mu_m(t_k)$ is the probability of the most likely cell sequence between $C_m(t_k)$ and $C_j(t_f)$, $\psi_m(t_k)$ is the index of the most likely next cell from $C_m(t_k)$, and $q^* = [q_i^*(t_s), q_j^*(t_1), \ldots, q_j^*(t_f)]$ are the indexes of the most likely cell sequence between $C_i(t_s)$ and $C_j(t_f)$. For the discussion of subsequent sections, let $\mathrm{MLOP}(\pi, \mathbf{A}(t_s), \ldots, \mathbf{A}(t_{f-l}), \Delta(j))$ where $\Delta(j) = [\delta_{1j} \ldots, \delta_{Nj}]$, denote the most likely path traveled by the odor between $C_i(t_s)$ and $C_j(t_f)$.

Note that either the forward or the backward VA would require that all $\mathbf{A}(t_i)$ be available for $i \in [t_s, t_f)$. For the method given in Appendix B, this only requires that $\{u_x(t_i), u_y(t_i)\}$ be stored. Note also that the algorithm implicitly assumes a known starting time. Because the starting time is not known, one approach is to use the Backward VA and to choose $t_s = \max_{t_k}(\mu_i(t_k))$ where $\mu_i(t_k)$ is defined by (17).

3) Most Likely Odor Detection Path From C_j: This subsection defines an algorithm to calculate the connected cell sequence between C_i and C_j at the present time that is most likely to detect odor, for the given model λ. Since $\alpha(t_n, t_o)$ is a map of the likely plume locations, the algorithm for finding a contiguous cell sequence between C_i and C_j that maximizes the probability of detection is as follows:

1) **Initialize:** $\psi_m(0) = 0$ and $\mu_m(0) = \delta_{mj}\alpha_m(t_f, t_0), 1 \leq m \leq N$ where δ_{mj} is the Kronecker Delta.
2) **Recursion:**

$$\mu_m(k+1) = \max_{1 \leq n \leq N} [\alpha_m(t_f, t_0) Ne_n(m)\mu_n(k)] \qquad 1 \leq n \leq N, 0 \leq k \leq s-1$$
$$\psi_m(k+1) = \arg \max_{1 \leq n \leq N} [\alpha_m(t_f, t_0) Ne_n(m)\mu_n(k)].$$

3) **Termination:**

$$q_i^*(s) = i, \quad q_i^*(k) = \psi_{q_i^*(k-1)}(k-1)$$
$$\mu_m(s) = \max_{1 \leq n \leq N} [\pi_m \alpha_m(t_f, t_0) Ne_n(m)\mu_n(s+1)]$$

where $Ne_n(m)$ is a neighbors function such that

$$Ne_n(m) = \begin{cases} 1, & \text{if } C_n \text{ is a neighbor of } C_m \\ 0, & \text{otherwise} \end{cases}$$

$\mu_m(k)$ is a likelihood function proportional to the probability of detecting odor in each cell along the most likely odor detection k step cell sequence between C_j and C_m at t_f, $\psi_m(k)$ is the index of the most likely previous cell to $C_m(k)$, s is the number of cells in the sequence and $q^* = [q_i^*(s), q_i^*(s-1), \ldots, q_i^*(0)]$ are the indexes of the most likely odor detection cell sequence between C_i and C_j. The neighbors function is straightforward to define. One approach is given in Appendix I.

The logic of this algorithm is as follows. Since $\mu_m(k)$ is proportional to the probability of odor detection along the k step cell sequence between C_m and C_j that is most likely to detect odor, the vector $\mu(0)$ is initialized to correctly represent the fact that the only zero step sequence must start and end in C_j. For $k > 0$, $\mu_m(k)$ is updated based on the likelihood of detection in C_m and the likelihood of detection in all cells along the $k-1$ step sequences to the neighbors of C_m. Note that if there is no k step cell sequence between C_i and C_j, then $\mu_m(k) = 0$.

V. Examples

This section presents examples of the application of the algorithms that are contained in the body of this paper. In all the examples, the search region is a rectangle defined by $x \in [0, 100]$ m and $y \in [-50, 50]$ m. The cellular subdivision of this rectangle uses $m = 40$, $n = 25$ so that $N = 1000$. For each of the example figures, the source is located at $(x, y) = (20, 0)$ m, which is in C_{488} (i.e., column $= 8$, row $= 13$).

Figs. 5–8 show the coordinates of each corner in the corresponding corner. The map is computed over the entire region for each figure. The search area that is of interest is the smaller rectangle indicated by the dashed line. The regular grid of arrows indicate the local flow velocity at the tail of the arrow at the time the plot was generated. The plume resulting from a continual release of odor, turbulent diffusion, and advection by the temporally and spatially varying fluid flow is the grey-scale meandering path of circular filaments that begins at $(x, y) = (20, 0)$ m. The plume simulation model is described in [13].

For Figs. 5 and 6, the flow field is defined by the simulation model and varies with both space and time as a function of time-varying boundary conditions. Fig. 5 shows the result of calculating $\alpha(65, 0)$, plotted as a grey-scale map,[2] using the flow

[2]Note that $\alpha(65, 0)$ is a vector in $\Re^N$ where $N = mn$. The map is produced by converting $\alpha(65, 0)$ to an $n \times m$ matrix and coloring each cell according to the value of the corresponding matrix element.

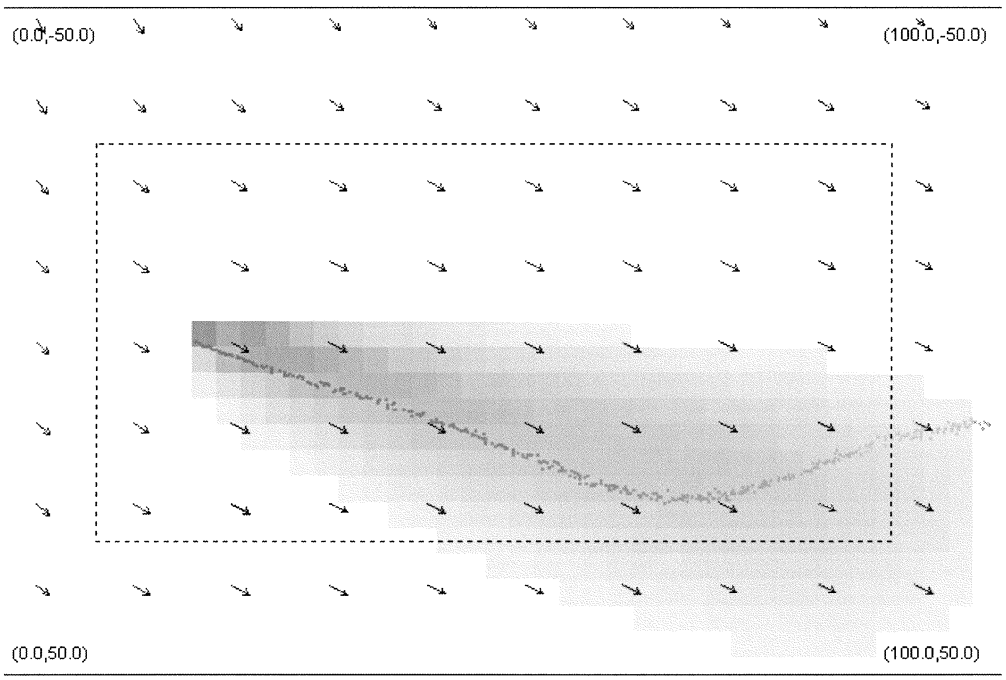


Fig. 5. Plume map $\alpha(65,0)$ from the forward algorithm assuming the source is in C_{488}. The array of arrows indicates the local flow velocity at the tail of the arrow. The coordinates of each corner are indicated in each corner. The dashed rectangle indicates the desired search area. The array of grey-scale rectangles indicates the size of $\alpha(65,0)$ in each cell, where darker cells have higher probability of containing the plume. The grey-scale patchy trail indicates the simulated odor concentration as a function of position (i.e., the plume). The plume shape is time-varying as determined by the advection of the time-varying flow field.

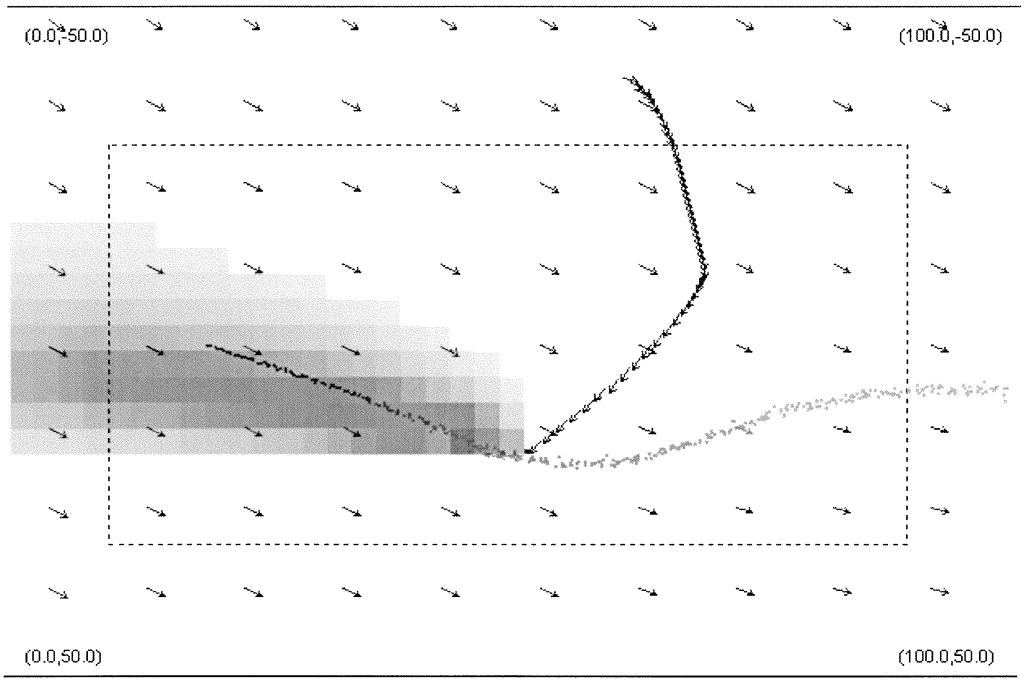


Fig. 6. Map representation of $\bar{\beta}_{619}(54,0)$ from the backward algorithm for odor detection occurring in $C_{619}(54)$. The array of arrows indicates the local flow velocity at the tail of the arrow. The coordinates of each corner are indicated in each corner. The dashed rectangle indicates the desired search area. The array of grey-scale rectangles indicates the size of $\bar{\beta}_{619}(54,0)$ in each cell, where darker cells have higher probability of transitioning detectable odor to the cell containing the vehicle at the time of the calculation. The grey-scale patchy trail indicates the simulated odor concentration as a function of position (i.e., the plume). The plume shape is time-varying as determined by the advection of the time-varying flow field. The trail of dark arrows moving from near the top edge down toward the plume indicates the trajectory that the vehicle followed. The initial vehicle position at $t = 0$ was in C_{64}.

velocity measured at each time-varying vehicle location. For the computation

$$\pi_i = \begin{cases} 1, & \text{for } i = k \\ 0, & \text{otherwise} \end{cases}$$

where $k = 488$ (i.e., $i = 8$, $j = 13$) is the cell index contain the true source location. This choice of k allows the plume likelihood predicted by α to be directly compared with the actual plume. The plume likelihood map is maximum directly down flow of the source. The plume likelihood map decreases rapidly

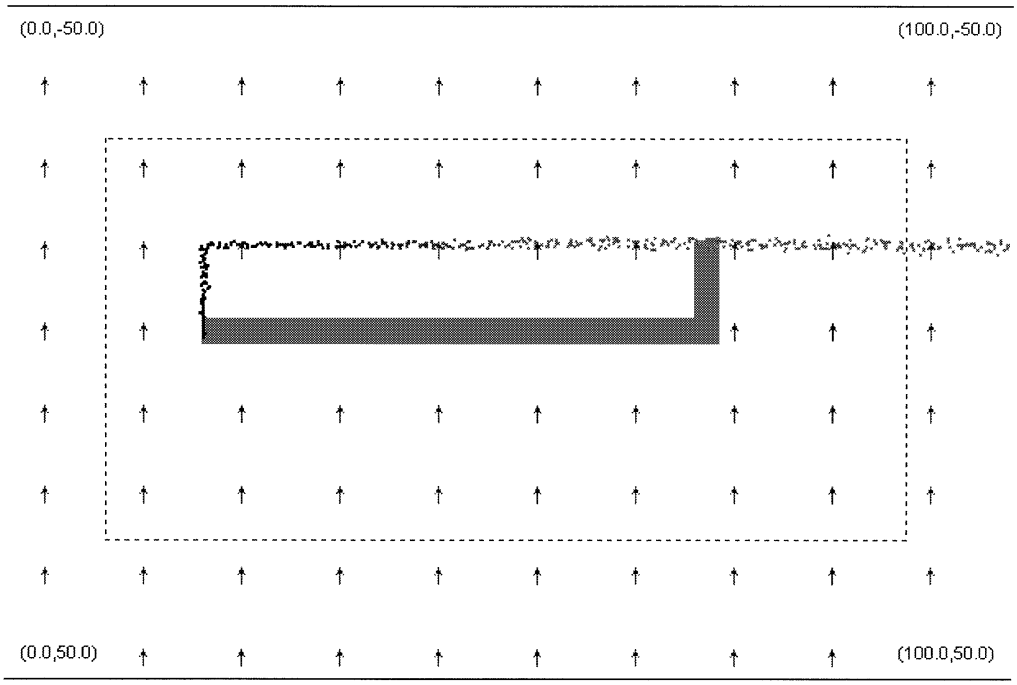


Fig. 7. Most likely path traveled by the odor between $C_{488}(0)$ and $C_{388}(44)$ shown by the grey-scale rectangles. The array of arrows indicates the local flow velocity at the tail of the arrow (at $t = 44$ s). To produce this graphic, the flow was artificially forced to be uniform over the region and to be 1 m/s in the x direction for till $t = 30$ s. For $t > 30$ s, the flow was forced to be uniform over the region and to be 1 m/s in the y direction. The coordinates of each corner are indicated in each corner. The dashed rectangle indicates the desired search area. The grey-scale patchy trail indicates the simulated odor concentration as a function of position (i.e., the plume). The plume shape is time-varying as determined by the advection of the time-varying flow field.

in the crossflow direction and more slowly in the downflow direction. The spread of the likelihood map increases with the downflow distance from C_k. All of these features are physically reasonable. If k were selected differently, then the plume likelihood map shape would not change, but its overlay on the region would be shifted to start at C_k. If π was selected to have more than one nonzero element, the algorithm is still valid without change. The resulting plume map would effectively be the appropriately scaled superposition of each separate source.

Fig. 6 shows a vehicle trajectory (The vehicle trajectory is indicated by the trail of arrows starting in C_{64} (i.e., $i = 24$, $j = 2$) at $t = 0$. The direction of each arrow indicates the vehicle heading). and a grey-scale map of $\bar{\beta}_k(54, 0)$ where $k = 619$ is defined by the cell containing the vehicle at the time the odor is detected. The map of $\bar{\beta}$ has its maximum immediately upflow of the vehicle locations. The map of $\bar{\beta}$ decreases rapidly in the crossflow directions and more slowly in the upflow direction. The map spreads out as it proceeds farther upflow.

For Figs. 7 and 8, we (artificially) imposed a uniform flow field over the entire region so that the validity of the resulting paths can be clearly observed. This uniform flow field for Figs. 7 and 8 is defined by

$$(u_x, u_y) = \begin{cases} (1,0), & \text{for } t \leq 30\,\text{s} \\ (0,1), & \text{otherwise.} \end{cases} \tag{18}$$

Fig. 7 shows the most likely path traversed by the odor between $C_{488}(0)$ and $C_{388}(44)$. Fig. 8 shows the 25 cell path between C_{388} and C_{488} at time t = 44 s that is most likely to detect odor. Note that these two paths are distinct. The algorithms presented in the body of this paper are valid for any record of flow velocities. For this example, we purposefully enforced the uniform flow field defined in (18) to allow the reader to easily verify the two paths that are shown.

VI. Conclusion

The algorithms presented herein were based on HMMs. Algorithms are presented for: 1) determining which cells are likely to contain detectable odor based on measured flow information and an assumed source probability vector; 2) determining which cells are likely to have resulted in odor at a point where it was detected (or not detected) based on measure flow data; 3) estimating a source probability vector; 4) determining the most likely path that odor took from an assumed source location to a cell that is of interest; and 5) determining the path of a given length between two given locations that is most likely to encounter odor. This path is interesting, because detection events produce the largest change to the source probability vector π, which is the only unknown portion of the HMM λ.

The algorithm given in Appendix II for computing $\mathbf{A}$ assumes that the flow velocity vector is spatially invariant. This assumption is not true, but is necessary based on the one vehicle assumption. The negative effects of this assumption will be significant if the search area contains significant terrain features that locally affect the flow or if the temporal variations of the flow are rapid enough that their propagation across the search region should be addressed. The temporal effects can be alleviated by choice of the search time for appropriate environmental conditions. The effect of this spatial invariance assumption is also decreased by the fact that π is estimated online based on

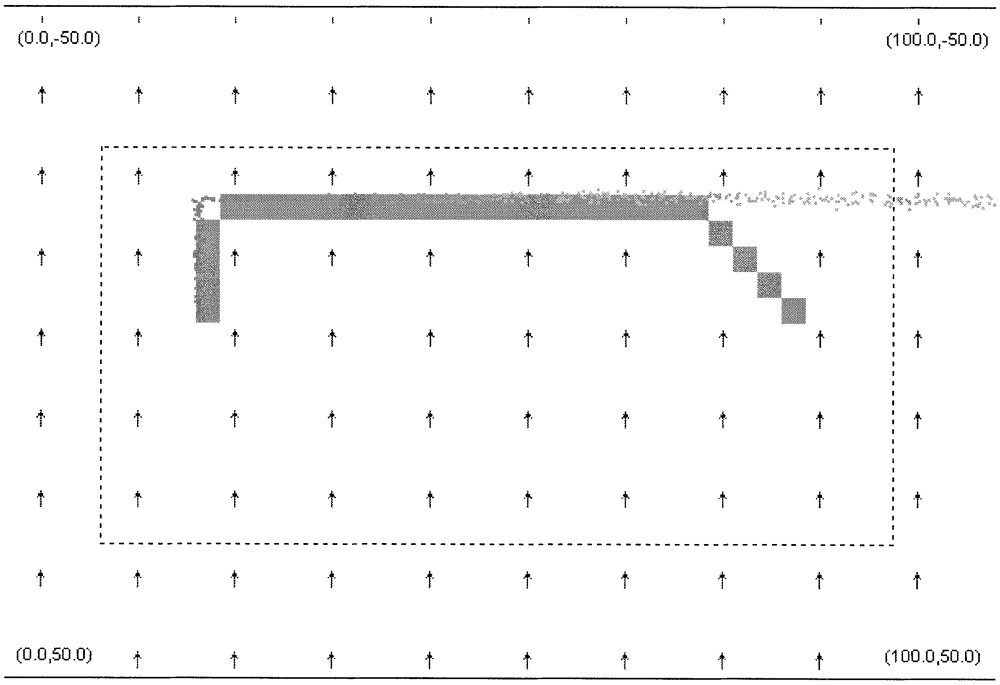


Fig. 8. Twenty-five step path between C_{388} and C_{471} that is most likely to detect odor at time $t = 48$ is shown by the grey-scale rectangles. The array of arrows indicates the local flow velocity at the tail of the arrow (at $t = 48$ s). To produce this graphic, the flow was artificially forced to be uniform over the region and to be 1 m/s in the x direction for till $t = 30$ s. For $t > 30$ s, the flow was forced to be uniform over the region and to be 1 m/s in the y direction. The coordinates of each corner are indicated in each corner. The dashed rectangle indicates the desired search area. The grey-scale patchy trail indicates the simulated odor concentration as a function of position (i.e., the plume). The plume shape is time-varying as determined by the advection of the time-varying flow field.

detection events. Because β is largest near the location of the detection event, the largest changes to π are near this location.

Note that the magnitude of the sensed chemical has not played a role in the derivations thus far. This is beneficial, because the accuracy of the $c(t)$ sensor is not critical. A binary chemical detector is sufficient. The approach might be improved by incorporating information about the magnitude of the sensed chemical. To date, this approach has not been pursued. Challenges to incorporating the magnitude of $c(t)$ include the fact that the source strength is not known and the fact that the sensor is not necessarily detecting the peak concentration of the parcel of odor in its vicinity.

Future work is still necessary to estimate the appropriate duration of the backward integration. Several approaches are possible. The fluid dynamics literature [36] provides methods for estimating the downflow distance from the source based on characteristics of the measured chemical. Alternatively, using the hidden Markov approaches described herein, the following ideas are of interest.

1) Propagate both the most likely odor path (MLOP) and the most likely detection path (MLDP) backward from the vehicle location. Find the points at which they intersect. Each intersection point is an estimate of the source location.
2) Let $c_k = \alpha_k \bar{\beta}_k$. The set $S = \{C_k \,|\, c_k > \tau\}$, where τ is a threshold, defines a set of points likely to be on both the MLDP and MLOP.

Processing of such sets of cells could provide an alternative means to estimate the SLIM represented by π. Such approaches work better when the flow varies significantly; however, there may be many points of intersection when the flow velocity is nearly constant.

One method for computation of the matrix $\mathbf{A}$ is given in Appendix II. This approach yields a $\mathbf{A}$ that is very sparse, containing only nine distinct nonzero elements. This approach, running on a 300–MHz computer, implements all four of the maps in the example section, a vehicle simulation, an environment simulation, and a planner in better than realtime. The main assumptions of that definition of $\mathbf{A}$ are that the flow is uniform over the region and that $|u|dt$ is smaller than the cell length. Many alternative approaches to calculate $\mathbf{A}$ can be constructed. For example, $\mathbf{A}$ can be generalized to account for uncertainty in cell transitions due to the temporal variation in the flow over each sample period.

This paper has only addressed the mapping portion of the overall problem. A planner is also required to determine the maneuvers that the vehicle should perform. The quality of the map will be strongly affected by the decisions of the planner. The planner must also address various mission objectives, energy and safety constraints. The planner used in the examples section commands the vehicle to enter the neighboring cell that presently has the highest probability of detecting odor (i.e., the 1-step MLDP). The planner and map interact since both recompute at 1 Hz. The resulting vehicle trajectories move across the flow while finding the plume and up the flow following odor detection. These characteristics are similar to those exhibited by various biological entities [24]. Starting from a random location in a 100×100 m search area, the vehicle typically locates the source in less than 300 s using a velocity of 1 m/s.

TABLE IV
NOTATION SUMMARY

$\alpha_k(t_f, t_s)$	The probability of there being detectable odor in C_k at time t_f due to the continual release of odor for $t \in [t_s, t_f]$.
$\bar{\alpha}_k(t_f, t_s)$	The probability of detectable odor in C_k at time t_f due to the one time release of odor at t_s.
$\beta_{ij}(t_f, t_s)$	The probability of odor released in $C_i(t_s)$ transitioning to $C_j(t_f)$
$\bar{\beta}_{ij}(t_f, t_s)$	The probability of odor released for $t \in [t_s, t_f]$ being detectable in $C_j(t_f)$
$\delta_t(i)$	Probability of the most likely cell sequence to $C_i(t)$
$\lambda = [\pi, \mathbf{A}(t), b]$	Hidden Markov model
π_k	Probability that there is an odor source in cell k
$\psi_t(j)$	The index of the most like cell transitioning to $C_j(t)$
μ	Probability of detecting odor given that there is detectable odor in the bin
$a_{kl}(t_i)$	Probability of transition of detectable odor from $C_k(t_i)$ to $C_l(t_{i+1})$
b_k	Probability of detecting odor in cell k given that there is detectable odor in cell k
$c(p(t_i))$	Concentration at p at t_i.
m	Number of x-axis subdivisions of the search area
n	Number of y-axis subdivisions of the search area
$p_v(t_i)$	Vehicle location at time t_i.
q_j^*	Most likely cell sequence to $C_j(T)$.
t_i	Time of the i-th set of measurements.
$\mathbf{u}(t_i) = (u_x(t_i), u_y(t_i))$	Flow velocity vector at time t_i .
$\mathbf{A}(t_i) = [a_{kl}(t_i)]$	Matrix of transition probabilities, $k,\ l \in [0, N]$
$\bar{\mathbf{A}}(t_i) = [a_{kl}(t_i)]$	Matrix of transition probabilities, $k,\ l \in [1, N]$
$C(p_v(t_i))$	The vehicle cell at time t_i.
$L_x(t) = \frac{X(t)}{m}$	The width of each cell at time t
$L_y(t) = \frac{Y(t)}{n}$	The length of each cell at time t
M	Number of sensor measurements per computation interval.
$N = n\ m$	Number of cells covering the region
$X = \bar{x} - \underline{x}$	The width of the searching area
$Y = \bar{y} - \underline{y}$	The length of the searching area

APPENDIX I
NEIGHBORS FUNCTION

For the algorithms used in the main body of this paper, it is useful to have a function that returns a vector k^* containing the indexes of the cells adjacent to cell k. This function is referred to as the neighbors function.

Let a rectangular region be divided into a grid of $n \times m$ cells. The cells will be indexed as $c_{ij}, i \in [1, m]$ and $j \in [1, n]$. For a nonedge cell, its (inclusive) neighboring cells are

$$\begin{matrix} c_{i-1,j-1} & c_{i-1,j} & c_{i-1,j+1} \\ c_{i,j-1} & c_{i,j} & c_{i,j+1} \\ c_{i+1,j-1} & c_{i+1,j} & c_{i+1,j+1}. \end{matrix}$$

If it is desirable to represent this set of (i, j) indexes by a vector with the index mapping $k = i + (j-1)m$, then in k-space the neighbors of cell k are

$$k^* = \{k-1-m, k-m, k+1-m, k-1 \\ k, k+1, k-1+m, k+m, k+1+m\}.$$

Each edge of the region must be treated separately, by replacing the (i, j) cell indexes that are outside of $[1, m] \times [1, n]$ with zero, since the zero cell represents the exterior region. For example, the upper edge (noncorner) cells of the region would have neighboring cells represented by

$$\begin{matrix} c_{i-1,0} & c_{i-1,j} & c_{i-1,j+1} \\ c_{i,0} & c_{i,j} & c_{i,j+1} \\ c_{i+1,0} & c_{i+1,j} & c_{i+1,j+1} \end{matrix}$$

and

$$k^* = \{0, 0, 0, k-1, k, k+1, k-1+m, k+m, k+1+m\}. \tag{19}$$

The neighbors along the other edges and corners are defined similarly.

APPENDIX II
TRANSPORT MAPPING

This appendix discusses one method to calculate the matrix $\mathbf{A}$, based on reasonable physical assumptions and the data available to the vehicle.

Let the rectangular search area be defined by the corners: $(\underline{x}, \underline{y}), (\underline{x}, \bar{y}), (\bar{x}, \bar{y})$, and $(\bar{x}, \underline{y})$. The length of the region in the x and y directions are $X = \bar{x} - \underline{x}$ and $Y = \bar{y} - \underline{y}$. The cell width in the x and y directions are $L_x = (X/m)$ and $L_y = (Y/n)$. Assume that dt is small enough so that $|\mathbf{u}|dt < (L_x, L_y)$ on a component-wise basis and $u(t)$ can be assumed constant over each time increment. The first portion of this assumption implies that material in any cell that is transported by the fluid flow moves a distance less than one cell width in the time dt.

The matrix $\mathbf{A}(t) = [a_{kl}(t)]$ represents the percentage amount (or probability) of material in $C_k(t)$ being transported to $C_l(t+1)$ by the fluid flow. $\mathbf{A}$ is a square matrix of dimension $mn + 1$. With the assumptions of the previous paragraph, the matrix $\mathbf{A}$ is sparse with at most nine nonzero elements per row. This fact greatly simplifies the HMM calculations and reduces the memory requirements (from $N \times N$ to $N \times 9$). Assuming that the fluid flow is spatially invariant, results in the conclusion that these nine nonzero values are the same in each row. This assumption is not strictly true, but is the best that can be done with the information available to the vehicle. This assumption greatly reduces both the computation and the memory requirements of the algorithm (from $N \times 9$ to 9).

The actual definition of $\mathbf{A}(t)$ based on $u(t)$ is tedious. For nonedge cells there are eight distinct cases to address. Edge cells require additional attention. Here, we include only the case of a nonedge cell where $u_x > 0$ and $u_y > 0$. In this case, odor in cell (i, j) can only transition to cells $c_{i,j}, c_{i,j+1}, c_{i+1,j}$, and $c_{i+1,j+1}$. The probability of transition of detectable material to each of these cells is, respectively

$$\left(1 - \frac{u_x dt}{L_x}\right)\left(1 - \frac{u_y dt}{L_y}\right), \quad \left(1 - \frac{u_x dt}{L_x}\right)\left(\frac{u_y dt}{L_y}\right)$$
$$\left(\frac{u_x dt}{L_x}\right)\left(1 - \frac{u_y dt}{L_y}\right), \quad \left(\frac{u_x dt}{L_x}\right)\left(\frac{u_y dt}{L_y}\right).$$

The remaining elements of this row of the $\mathbf{A}$ matrix are zero. Note that this definition of $\mathbf{A}$ has the required property that each row of $\mathbf{A}$ sums to one. Each row of $\mathbf{A}$ can be efficiently computed using the neighbors function given in Appendix I.

APPENDIX III
NOTATION

Table IV summarizes the notation used throughout the paper.

ACKNOWLEDGMENT

The ideas presented here were inspired by the various meetings related to the CPT and CSME programs and especially by interactions with R. Cardé and J. Murlis.

REFERENCES

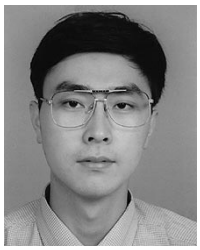
[1] J. Atema, "Chemical signals in the marine environment: Dispersal, detection, and temporal signal analysis," in *Proc. Nat. Acad. Sci.*, vol. 92, 1995, pp. 62–66.
[2] D. E. Aylor, "Estimating peak concentrations of pheromones in the forest," in *Perspectives in Forest Entomology*, J. F. Anderson and M. K. Kaya, Eds. New York: Academic, 1976, pp. 177–188.
[3] J. Basil, "Lobster orientation in turbulent odor plumes: Simultaneuos measuremetns of tracking behavior and temporal odor patterns," *Biol. Bull.*, vol. 187, pp. 272–273, 1994.
[4] J. H. Belanger and M. A. Willis, "Adaptive control of odor-guided location: Behavioral flexibility as an antidote to environmental unpredictability," *Adaptive Behavior*, vol. 4, pp. 217–253, 1998.
[5] H. C. Berg, "Bacterial microprocessing," in *Proc. Cold Springs Harbor Symp. Quantum Biology*, vol. 55, 1990, pp. 539–545.
[6] R. T. Cardé, "Odor plumes and odor-mediated flight in insects. In olfaction in mosquito-host interactions," in *Proc. CIBA Foundation Symp. 200*. New York, 1996, pp. 54–70.
[7] R. T. Cardé and A. Mafra-Neto, "Mechanisms of flight of male moths to pheromone," in *Insect Pheromone Research. New Directions*, R. T. Cardé and A. K. Minks, Eds. London, U.K.: Chapman & Hall, 1996, pp. 275–290.
[8] R. Deveza, D. Thiel, A. Russell, and A. Mackay-Sim, "Odor sensing for robot guidance," *Int. J. Robot. Res.*, vol. 13, pp. 232–239, 1994.
[9] D. V. Devine and J. Atema, "Function of chemoreceptor organs in spatial orientation of the lobster," *Homarus Americanus: Differences and Overlap*, vol. 163, pp. 144–153, 1982. Biol. Bull..
[10] D. B. Dusenbery, *Sensory Ecology: How Organizms Acquire and Respond to Information*. San Francisco, CA: Freeman, 1992.
[11] J. S. Elkinton, R. T. Cardé, and C. J. Mason, "Evaluation of time-average dospersion models for estimating pheromone concentration in a deciduous forest," *J. Chem. Ecol.*, vol. 10, pp. 1081–1108, 1984.
[12] J. S. Elkinton, C. Schal, T. Ono, and R. T. Cardé, "Pheromone puff trajectory and upwind flight of male gypsy moths in a forest," *Physiol. Entomology*, vol. 12, pp. 399–406, 1987.
[13] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. Cardé, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," Environmental Fluid Mechanics, vol. 2, pp. 143–169, Aug. 2002.
[14] G. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, 1973.
[15] F. W. Grasso, T. Consi, D. Mountain, and J. Atema, "Locating odor sources in turbulence with a lobster inspired robot," in *From Animals to Animals 4: Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, Eds. Cambridge, MA, 1996, pp. 104–112.
[16] F. W. Grasso, T. R. Consi, D. C. Mountain, and J. Atema, "Biomimetic robot lobster performs chemo-orientation inturbulence using a pair of spatially separated sensors: Progress and challenges," *Robot. Autom. Syst.*, vol. 30, pp. 115–131, 2000.
[17] F. W. Grasso, "Invertebrate-inspired sensory-motor systems and autonomous, olfactory-guided exploration," *Biol. Bull.*, vol. 200, pp. 160–168, 2001.
[18] A. D. Hassler and A. T. Scholz, *Olfactory Imprinting and Homing in Salmon*. New York: Springer-Verlag, 1983.
[19] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi, "Odor-source localization in the clean room by an autonomous mobile sensing system," *Sens. Actuators B, Chem.*, vol. 33, pp. 115–121, 1996.
[20] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata, "Plume-tracking robots: A new application of chemical sensors," *Biol. Bull.*, vol. 200, pp. 222–226, 2001.
[21] C. D. Jones, "On the structure of instantaneous plumes in the atmosphere," *J. Hazard. Mater.*, vol. 7, pp. 87–112, 1983.
[22] K.-E. Kaissling, "Pheromone-controller anemotaxis in moths," in *Orientation and Communication in Arthropods*, M. Lehrer, Ed. Cambridge, MA: Birkhaüser, 1997, pp. 343–374.

[23] Y. Kuwana, S. Nagasawa, I. Shimoyama, and R. Kanzaki, "Synthesis of the pheromone-oriented behavior of silkworm moths by a mobile robot with moth antennae as pheromone sensors," *Biosensors Bioelectronics*, vol. 14, pp. 195–202, 1999.
[24] W. Li, J. A. Farrell, and R. T. Cardé, "Tracking of fluid-advected odor plumes: Strategies inspired by insect orientation to pheromone," Adaptive Behavior, to be published.
[25] K. J. Lohmann, "How sea turtles navigate," *Sci. Amer.*, vol. 266, pp. 82–88, 1992.
[26] A. Mafra-Neto and R. T. Cardé, "Fine-scale structure of pheromone plumes modulates upwind orientation of flying moths," *Nature*, vol. 369, pp. 142–144, 1994.
[27] J. Murlis and C. D. Jones, "Fine scales structure of odor plumes in relation to insect orientation to distant pheromone and other attractant sources," *Physiol. Entomology*, vol. 6, pp. 71–86, 1981.
[28] J. Murlis, J. S. Elkinton, and R. T. Cardé, "Odor plumes and how insects use them," *Annu. Rev. Entomology*, vol. 37, pp. 505–532, 1992.
[29] K. R. Mylne, "Concentration fluctuation measurements in a plume dispersing in a stable surface layer," *Boundary-Layer Meteorology*, vol. 60, pp. 15–48, 1992.
[30] G. A. Nevitt, "Olfactory foraging by antartic proellariiform seabirds: Life at high Reynolds numbers," *Biol. Bull.*, vol. 198, pp. 245–253, Apr. 2000.
[31] A. B. Poritz, "Hidden Markov models: A guided tour," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 7–13, 1988.
[32] R. Preiss and E. Kramer, "Stabilization of altitude and speed in tethered flying gypsy moth males: Influence of $(+)$ and $(-)$-disparlure," *Physiol. Entomology*, vol. 8, pp. 55–68, 1983.
[33] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, 1989.
[34] P. Rau and N. L. Rau, "The sex attraction and rhythmic periodicity in the giant Saturniid moths," *Trans. Academy of Sciences of Saint Louis*, vol. 26, pp. 82–221, 1929.
[35] B. I. Shraiman and E. D. Siggia, "Scalar turbulence," *Nature*, vol. 405, pp. 639–646, 2000.
[36] M. T. Stacey, E. A. Cowen, T. M. Powell, E. Dobbins, S. G. Monismith, and J. R. Koseff, "Plume dispersion in a stratified, near-coastal flow: Measurements and modeling," *Continental Shelf Res.*, vol. 20, pp. 637–663, 2000.
[37] O. G. Sutton, "The problem of diffusion in the lower atmosphere," *Q. J. R. Meteorol. Soc.*, vol. 73, pp. 257–281, 1947.
[38] ——, *Micrometeorology*. New York: McGraw-Hill, 1953.
[39] N. J. Vickers, "Mechanisms of animal navigation in odor plumes," *Biol. Bull.*, vol. 198, pp. 203–212, 2000.
[40] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, 1967.
[41] D. R. Webster, S. Rahman, and L. P. Dasi, "On the usefulness of bilateral comparison to tracking turbulent chemical odor plumes," *Limnol. Oceanogr.*, vol. 46, pp. 1048–1053, 2001.
[42] M. J. Wiesburg and R. K. Zimmer-Faust, "Odor plumes and how blue crabs use them in finding prey," *J. Exp. Biol.*, vol. 197, pp. 349–375, 1994.
[43] R. K. Zimmer and C. A. Butman, "Chemical signaling processes in the marine environment," *Biol. Bull.*, vol. 198, pp. 168–187, 2000.

Jay A. Farrell (M'85-SM'98) received the B.S. degrees in physics and electrical engineering from Iowa State University, Ames, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1988 and 1989, respectively.

He joined the technical staff at the Charles Stark Draper Laboratory, Cambridge, MA, in 1989. Currently, he is a Professor of Electrical Engineering at the University of California, Riverside. He served as Chair of the Department of Electrical Engineering from 1998 to 2001. His current research interests are identification and online control for nonlinear systems, integrated GPS INS navigation, and artificial intelligence techniques for autonomous dynamic systems. He is the author of the book The Global Positioning System and Inertial Navigation, New York: McGraw-Hill, 1998 and over 85 additional technical publications.

Dr. Farrell received the Engineering Vice President's Best Technical Publication Award in 1990, and the Recognition Award for Outstanding Performance and Achievement in 1991 and 1993. He is a former Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS and IEEE TRANSACTIONS ON AUTOMATIC CONTROL.



Shuo Pang received the B.Eng. degree in electrical engineering from Harbin Engineering University, Harbin, China in 1997, and the M.S. degree in electrical engineering from the University of California, Riverside, CA in 2001, where he is currently pursuing the Ph.D. degree.

His current research interests include hybrid electrical vehicle energy management systems and artificial intelligence techniques for autonomous vehicle, e.g., autonomous vehicle chemical plume tracing, autonomous vehicle online mapping, and planning.

Wei Li received the B.S. and M.S. degrees in electrical engineering from the Northern Jiaotong University, Beijing, China, in 1982 and 1984, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Saarland, Saarland, Germany, in 1991.

He was a Faculty Member of Computer Science and Technology at Tsinghua University, Beijing, from 1993 to 2001. In 2001, he became an Associate Professor of Computer Science at California State University, Bakersfield, CA. His research interests are intelligent systems, robotics, fuzzy logic control and neural networks, multisensor fusion and integration, and graphical simulation.

Dr. Li received the 1995 National Award for Outstanding Postdoctoral Researcher in China and the 1996 Award for Outstanding Young Researcher at Tsinghua University. He was a Croucher Foundation Research Fellow at City University of Hong Kong (1996) and an Alexander von Humboldt Foundation Research Fellow at the Technical University of Braunschweig (1997–1998).